เครื่องมือ
สำหรับนักพัฒนา
ซอฟต์แวร์
(Tools for Software Developers)

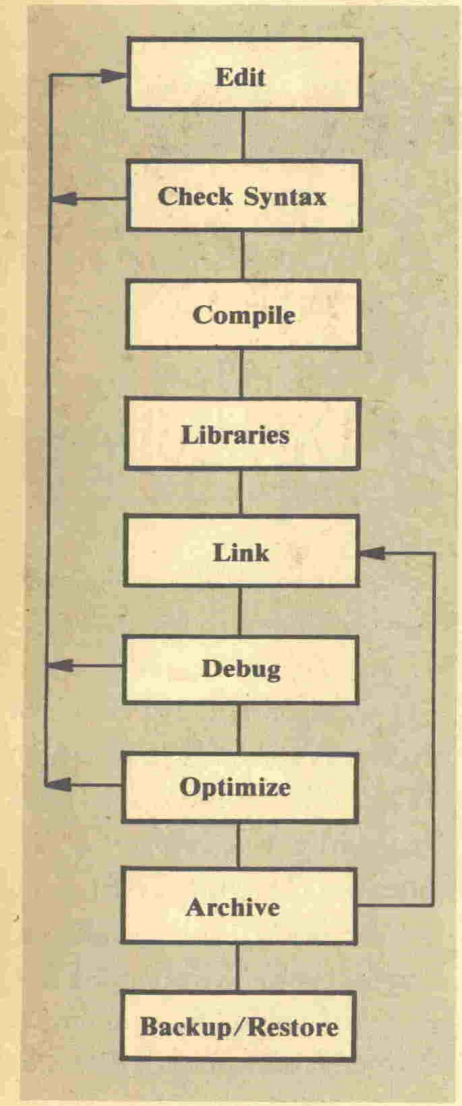
ชัยยงค์ วงศ์ชัยสุวัฒน์

หากเรานับอายุขัยของเครื่องไมโครคอมพิวเตอร์ ตั้งแต่รุ่นแอปเปิ้ล 8 บิต รุ่นไอบีเอ็ม พีซี, เอกซ์ที, เอที จนถึงปัจจุบันรุ่น PS/2 หรือรุ่น 386 จะเป็นเวลานานมากกว่า 10 ปีแล้ว ตลอดระยะเวลาที่ผ่านมา ซอฟต์แวร์ที่ใช้หรือพัฒนาก็เพิ่มความยุ่งยากซับซ้อนมากขึ้นด้วยเป็นเงาตามตัว ซึ่งหากจะเขียนซอฟต์แวร์ประยุกต์กับงานหนึ่ง ๆ ให้มีขีดความสามารถของการใช้งานสูงขึ้นและสะดวกขึ้นก็ต้องใช้เวลาในการพัฒนาค่อนข้างนาน เพื่อที่จะลดเวลาของการพัฒนาและเพิ่มขีดความสามารถของการใช้งาน จึงได้มีการพัฒนาเครื่องมือซึ่งก็คือโปรแกรมนั่นเองขึ้นมาช่วยนักโปรแกรมอีกต่อหนึ่งในการเขียน ทดสอบ และผลิตซอฟต์แวร์ประยุกต์ที่มีประสิทธิภาพดีที่สุด

เครื่องมือที่ช่วยนักโปรแกรมในการพัฒนาซอฟต์แวร์มีอยู่มากมายหลายชนิดด้วยกัน ในบทความนี้จะนำเสนอประเภทต่าง ๆ ของเครื่องมือเหล่านี้ (โดยไม่รวมถึงเครื่องมือที่นำมาใช้ประกอบซอฟต์แวร์สำเร็จรูปอื่นเช่น dBASE, LOTUS 123 ฯลฯ) เครื่องมือบางประเภทก็มีขายหรือใช้ในบ้านเราแล้วแต่บางประเภทก็ยังไม่มีในบ้านเรา ซึ่งหากใครเห็นประโยชน์ที่อาจจะได้จากเครื่องมือเหล่านี้ก็อาจหามาลองใช้ได้

เครื่องมือพัฒนาโปรแกรม

เมื่อพิจารณาขั้นตอนการพัฒนาซอฟต์แวร์ ดังรูปที่ 1 เริ่มจากการนำเข้า source code ของโปรแกรมไปจนกระทั่งถึงการเก็บในดิสก์ ในแต่ละขั้นตอนเราสามารถนำเครื่องมือมาช่วยในการทำงานได้ ซึ่งเครื่องมือแต่ละประเภทก็มีผู้ผลิตออกมาหลายราย ด้วยราคาขายที่แตกต่างกันตามขีดความสามารถของการทำงาน ต่อไปนี้ก็จะมาแยกดูแต่ละประเภท โดยไม่รวมถึงโปรแกรมของตัวแปลภาษา Compile และโปรแกรม Linkage & Loader



รูปที่ 1 ขั้นตอนการพัฒนาซอฟต์แวร์

ซอฟต์แวร์ Text Editor

การพัฒนาซอฟต์แวร์ เมื่อเราได้แนวความคิดของโปรแกรมแล้ว ขั้นตอนต่อไปก็จะเริ่มเขียนโปรแกรมแล้วจะนำเข้ารหัสด้วยภาษาคอมพิวเตอร์ ในตอนนี้นักโปรแกรมจำต้องอาศัยเครื่องมือคือโปรแกรมเอดิเตอร์มาช่วยเก็บ source code และเมื่อเก็บแล้วก็สามารถเรียกกลับมาแก้ไขภายหลังได้ ลักษณะของเอดิเตอร์นี้จะช่วยนักโปรแกรมนำเข้า source code ได้ง่ายและรวดเร็ว ซึ่งเอดิเตอร์ที่ดีที่สุดสำหรับนักโปรแกรมคนหนึ่ง อาจไม่ถูกใจสำหรับนักโปรแกรมผู้อื่นก็ได้ ทั้งนี้ขึ้นกับผู้ใช้จะพึงพอใจหรือคุ้นเคยกับระบบเดิมที่เคยใช้เพียงไร

ลักษณะของโปรแกรมเอดิเตอร์นี้จะแตกต่างกับโปรแกรมเวิร์ดโปรเซสซิ่งใน 2 ลักษณะคือ โปรแกรมเอดิเตอร์จะไม่เอื้ออำนวยหรือสนับสนุนฟังก์ชันเกี่ยวกับการพิมพ์ และไม่มีลักษณะพิเศษเกี่ยวกับการทำ foot note, หมายเลขหน้า, การทำตารางสารบัญ เหมือนโปร-แกรมเวิร์ดโปรเซสซิ่งทั่ว ๆ ไป แต่ในโปรแกรมเอดิเตอร์ที่เก็บข้อมูลที่เป็นเท็กซ์ของ source code นี้ จะมีลักษณะพิเศษอื่นมาทดแทน ตัวอย่างเช่น

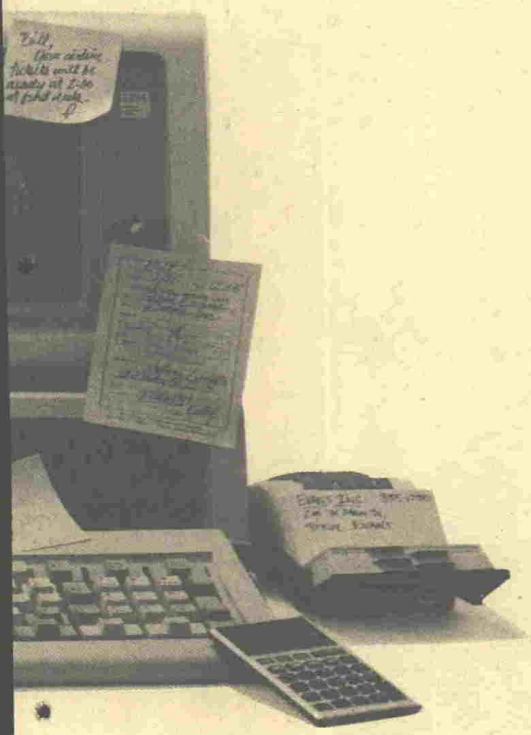
- *การกำหนดปุ่มป้อนของคีย์บอร์ดใหม่*
- *มีวิธีใช้ที่คล้ายคลึงกับเอดิเตอร์ที่ใช้บนเครื่องมินิหรือเมนเฟรม*
- *มี macro command เฉพาะสำหรับภาษาที่ใช้*
- *การจัดการบนไฟล์หลายไฟล์พร้อมกัน*
- *การทำงานแยกเป็น window*
- *การเรียก DOS Shell*
- *สามารถทำงานทั้ง foreground และ background*
- *ขนาดของไฟล์ไม่จำกัดจนถึงขีดสุดของเนื้อที่ว่างบนดิสก์*

- *การย่อหน้าอัตโนมัติ ตามลักษณะเฉพาะของภาษาที่ใช้*
- *มีการใช้ "garbage stack" มาเก็บข้อมูลที่ลบแล้วเรียกกลับคืนได้*

ตัวอย่างของเครื่องมือเอดิเตอร์ในกลุ่มนี้ได้แก่ Norton Editor ของ Peter Norton Computing, BRIEF ของ Solution Systems

ซอฟต์แวร์ตรวจสอบหลักไวยากรณ์ทางภาษาที่ใช้

เครื่องมือช่วยพัฒนานี้อาศัยหลักการเช่นเดียวกับ UNIX "lint" สำหรับภาษา C ซึ่งจะตรวจค้นข้อผิดพลาดที่หลุดลอดตัวแปลภาษา คอมพายเลอร์ หรือตัว debugger เครื่องมือนี้จะช่วยรายงานสิ่งไม่ถูกต้องมาให้ผู้ใช้จัดการแก้ไขภายหลัง ซอฟต์แวร์ประเภทนี้ส่วนใหญ่จะเป็นแบบที่สนับสนุนภาษา C ใช้ได้กับ



ภาษา C จากหลาย ๆ แหล่ง เช่น Microsoft, Mark Williams และ Lattice ตัวอย่างซอฟต์แวร์นี้ได้แก่ Pre-C ของ Phoenix Computer Product Corporation, PC-lint ของ Gimpel Software

ซอฟต์แวร์ไลบรารี่ของรูทีนใช้งาน

ซอฟต์แวร์ไลบรารี่ของรูทีนสำเร็จเรียกมาใช้ทันทีนี้ ปัจจุบันมีผู้ผลิตออกมาและใช้กันอย่างกว้างขวาง ซึ่งรูทีนต่าง ๆ ภายในมักจะประกอบด้วย

- *รูทีนจัดการจอภาพเกี่ยวกับการแสดงสีและภาพบนจอ*
- *รูทีนจัดการ window ของ overlapping, tiled, exploding*
- *รูทีนจัดการ Menu ของ pull-down, pop-up และแบบของ Lotus*
- *รูทีนจัดการ Error ของ ดอส, โปรแกรมและผู้ใช้เอง*
- *รูทีนต่อเชื่อมกับดอส ของการจัดการไฟล์, ดิสค์*
- *รูทีนจัดการคีย์บอร์ด*
- *รูทีนเกี่ยวกับการนำเข้าและแก้ไขข้อมูล*
- *รูทีนขอความช่วยเหลือ*
- *รูทีนช่วยกำหนดรูปแบบข้อมูลนำเข้า*

จากที่แสดงไว้ข้างต้น จะเห็นว่ารูทีนต่าง ๆ มีมากมายหลายหมวด จึงเป็นเครื่องมือที่ดีมากสำหรับนักโปรแกรมไม่ต้องใช้เวลาไปเขียน routine code เสียเองทั้งหมด เมื่อต้องการรูทีนใด ก็เพียงระบุชื่อของฟังก์ชันหรือรูทีน ช่วยให้ประหยัดเวลาเป็นอย่างมาก ตัวอย่างซอฟต์แวร์กลุ่มนี้ได้แก่ C TOOL PLUS ของ Blaise Computing และ Greenleaf Libraries, C-Worthy Interface Library ของ Solution Systems เป็นต้น

ซอฟต์แวร์ตรวจสอบข้อผิดพลาดของโปรแกรม (Debugger)

ตามปกติแล้ว โปรแกรมตรวจสอบข้อผิดพลาดของโปรแกรมจะรวมมาอยู่ในชุดของโปรแกรมตัวแปลภาษาแล้ว คือในเวลาเราคอมพายล์โปรแกรม เราสามารถติดตามดูข้อมูลต่าง ๆ ที่กำหนดขึ้นของตัวแปร ข้อผิดพลาดของโปรแกรมเราจะตรวจสอบค่า symbol ต่าง ๆ ที่กำหนดทั้งแบบ local และ global ได้ทุกขั้นของคำสั่งในระดับของภาษา source และภาษาแอสแซมบลี ลักษณะเช่นนี้จะพบใน CodeView ของ Microsoft หรือใน wvideo ของ WATCOM เป็นต้น

มีเครื่องมือบางประเภทจะเป็นโปรแกรมแยกต่างหาก โดยจะเป็น symbolic debugger ที่ทำงานร่วมกับภาษาทางคอมพิวเตอร์ของ IBM หรือ Microsoft ได้ เป็นโปรแกรมที่ใช้ง่าย ลักษณะ menu-driven หลาย window ที่ดึงตารางสัญลักษณ์ (symbol table) ได้จากโปรแกรม LINK ของ Microsoft หรือ PLINK 86 Plus ของ Phoenix Computer เครื่องมือนี้จะแสดง Source, การแปลทางแอสแซมบลี, เนื้อที่ stack, เนื้อที่ข้อมูลและตำแหน่ง break ได้พร้อม ๆ กัน ซึ่งจะเป็นประโยชน์มากต่อผู้พัฒนาซอฟต์แวร์ประยุกต์ ตัวอย่างของเครื่องมือนี้ได้แก่ PFIX86 Plus ของ Phoenix Computer Product, Periscope Products

ซอฟต์แวร์ช่วยปรับปรุงโปรแกรมให้ดีที่สุด

เครื่องมือนี้จะช่วยในการปรับแต่งซอฟต์แวร์ที่เขียนขึ้น โดยมันจะช่วยชี้บอกส่วนของโปรแกรมที่ไม่มีประสิทธิภาพ ที่ต้องเขียนใหม่เพื่อให้การทำงานได้เร็วขึ้น เครื่องมือนี้จะช่วยในการวิเคราะห์โปรแกรมในระหว่างที่โปรแกรมกำลังรัน แล้วบอกว่าทำงานอยู่รูทีนไหน ถูกเรียกจากจุดใด รูทีนนี้ถูกเรียกคำนวณกี่ครั้ง ครั้งละนานเท่าไร และมีจำนวนคำสั่งภายในรูทีนเท่าไร นอกจากนี้ยังสามารถใช้ข้อมูลของตารางสัญลักษณ์ (Symbol Table) มาวิเคราะห์โอเวอร์เลย์และการอินเตอร์รัพต์ได้ด้วย ตัวอย่างของเครื่องมือนี้ได้แก่ PFINISH ของ Phoenix Computer Product

ซอฟต์แวร์แบบยูทิลิตี้ "make" ใน UNIX

เครื่องมือหรือซอฟต์แวร์ช่วยพัฒนาข้างต้น จัดเป็นเครื่องมือที่นำมาใช้ในการพัฒนาโปรแกรมโดยเฉพาะ ซึ่งแต่เดิมจะอยู่ในสิ่งแวดล้อมของโปรแกรม

จัดระบบงานเอ็มเอสดอส แต่ในปัจจุบันได้เริ่มมีการนำได้หลักการของโปรแกรมจัดระบบงาน UNIX มาประยุกต์ใช้ ตัวอย่างอันหนึ่งก็คือ โปรแกรมยูทิลิตี้ "make" ของ UNIX โดยยูทิลิตี้นี้จะให้มีการสร้างไฟล์ชื่อ makefile ซึ่งเป็นการเก็บชุดของกฎการคอมพายล์และการลิงค์ของไฟล์ต่าง ๆ

ยูทิลิตี้ make นี้ไม่เพียงแต่แค่คอมพายล์และลิงค์เท่านั้น มันยังจะมีการตรวจสอบวันเวลาของไฟล์ด้วย เช่น ถ้า main.C แก้ไขสุดท้ายวันที่ 7/22/88 แต่ main. obj มีวันที่ 7/20/88 ในกรณีนี้เครื่องมือ make นี้จะทำการคอมพายล์ main.C เสียใหม่ก่อนนำไปลิงค์ตัวอย่างของเครื่องมือนี้ได้แก่ Poly Make ของ Polytron, LMK ของ Lattice, Pmaker ของ Phoenix Computer Product

ซอฟต์แวร์ควบคุมเวอร์ชัน

เครื่องมือนี้ค่อนข้างจะเป็นของใหม่มากอันหนึ่ง ที่เป็นซอฟต์แวร์มาช่วยจัดการควบคุมการปรับปรุงแก้ไข source code ของโปรแกรมประยุกต์ที่เขียนขึ้นเป็นการ update เวอร์ชันของซอฟต์แวร์ที่อาจมีคนมาช่วยกันเขียนหลาย ๆ คน โดยไม่เกิดการสับสนวุ่นวาย

ในปัจจุบัน ซอฟต์แวร์ที่พัฒนาขึ้นจะมีความซับซ้อนเพิ่มขึ้นเรื่อย ๆ ขนาดของโปรแกรมก็ใหญ่ขึ้นเรื่อย ๆ เช่นกัน ปัญหาใหญ่ของการพัฒนานี้อยู่ที่การจัดการควบคุมโครงการว่า ใครทำอะไรอยู่และทำไว้ถึงระดับไหนแล้ว เมื่อมีใครมาทำงานต่อจะสามารถหยิบ source file ที่แก้ไขสุดท้ายขึ้นมาปรับปรุงทำต่อได้ถูก ไม่หยิบเวอร์ชันเก่าผิดขึ้นมา และจะเลวร้ายมากถ้าเกิดการแก้ไขปรับปรุง source file เดียวกันไปพร้อม ๆ กัน จึงได้มีการพัฒนาเครื่องมือขึ้นมาช่วยจัดการ

เรียกว่า Version Control System เขียนย่อเป็น VCS

ซอฟต์แวร์ VCS ที่ดีควรจะมีความสามารถจัดการ การพัฒนา source code ของโปรแกรมต่อไปนี้ได้

ก. ติดตามการปรับปรุง source module ทุกครั้ง พร้อมกับมีการเก็บคำอธิบายประกอบการปรับปรุง ที่สามารถเรียกมาดูได้ทุกเวลา

ข. มีขีดจำกัดของการแก้ไขปรับปรุง source code เดียวกันพร้อม ๆ กันจากนักโปรแกรมสองคน

ค. ในกรณีที่เกิดการแก้ไข source code พร้อม ๆ กัน โดยอุบัติเหตุหรือความจงใจก็ตาม ก็สามารถนำมาประสานรวมกันได้

ง. มีระบบจัดเก็บและเรียกดูข้อมูลของผู้เขียนโปรแกรมว่าทำอะไรไว้บ้างและทำไว้เมื่อไร

อย่างไรก็ตาม ซอฟต์แวร์ VCS ก็ไม่ได้จำเพาะว่าจะนำมาใช้จัดการการเขียนโปรแกรมอย่างเดียว เรามักนำไปประยุกต์ใช้งานเอกสารที่เกี่ยวพันกับเวลา เช่น ราคาขาย, สัญญาต่าง ๆ เป็นต้น

เครื่องมือ VCS นี้จะใช้หลักการ "reverse delta storage" ที่ช่วยประหยัดเนื้อที่ของดิสค์ และช่วยในการเรียกเวอร์ชันของ source code ออกมาอย่างรวดเร็ว source code แรกดั้งเดิมจะถูกเก็บไว้ในที่เก็บ เมื่อมีการปรับปรุงแก้ไข source VCS ไม่ได้เก็บข้อมูลของการปรับปรุงแก้ไขใหม่ทั้งหมด แต่ไฟล์เดิมและตัวใหม่จะมีการเปรียบเทียบกันในลักษณะบรรทัดต่อบรรทัด แล้วจึงบันทึก

เฉพาะส่วนที่แตกต่างกันเรียกว่า "delta" ในส่วนที่เหมือนกันก็จะไม่มีการบันทึก ด้วยวิธีการนี้จะเป็นการประหยัดเนื้อที่ การเก็บข้อมูลที่ปรับปรุงแก้ไขได้มาก ซึ่ง "delta" ที่เก็บไว้ก็สามารถดูข้อมูลเดินหน้า (forward) หรือถอยกลับ (backward) ได้ เมื่อเราอยู่ที่ source code เวอร์ชันหนึ่ง ถ้าต้องการดู source code เวอร์ชันเก่าก็ใช้ delta ถอยกลับมาประกอบ และถ้าต้องการดู source code เวอร์ชันใหม่กว่าก็ใช้ delta เดินหน้ามาประกอบ เป็นต้น

เครื่องมือนี้อาจแบ่งเป็นหลายระดับตามลักษณะของการใช้งาน คือแบ่งเป็นแบบ Personal สำหรับนักโปรแกรมทีละคน แบบ Corporate สำหรับ source code ขนาดใหญ่หลาย ๆ โมดูลที่ซับซ้อน สำหรับนักโปรแกรมทำงานได้หลาย ๆ คนพร้อมกัน และแบบ Networks ที่ใช้เชื่อมโยงเป็นเครือข่าย มีวิธีจัดระดับความปลอดภัยของไฟล์

ตัวอย่างของซอฟต์แวร์ VCS ที่มีขายในต่างประเทศ ได้แก่ PVCS ของ Polytron, RCS ของ Mortice Kern Systems, SVM ของ Seidi Computer Engineering, TLIB ของ Burton เป็นต้น

ซอฟต์แวร์ภาษา AWK

ภาษา AWK เป็นภาษาที่พัฒนาสำหรับระบบ UNIX โดย Alfred Aho, Richard Weinberger และ Brian Kernigham ของ Bell Labs (ชื่อของบุคคลทั้งสามมารวมกันเป็นชื่อของภาษาใหม่นี้) เมื่อปี ค.ศ. 1977 ภาษานี้จะเป็นภาษาแบบ pattern matching และ C-like action ที่เขียนด้วยโปรแกรมสั้น ๆ แต่สามารถทำงานจัดการไฟล์ เพื่อการเรียกมา, การเปลี่ยนแปลงและการลดย่อได้ดี

โปรแกรมภาษา AWK นี้จะประกอบด้วยชุดของ patterns และ actions ที่จะต้องตรวจดูข้อมูลที่นำเข้าตรงกับ pattern และถ้าพบจะทำอะไรตามที่ระบุในส่วนของ action ที่คล้ายกับคำสั่งของภาษา C โปรแกรม AWK จะค้นในไฟล์เพื่อหาบรรทัดที่เหมือน pattern ซึ่ง pattern นี้จะเป็นการเลือกบรรทัดด้วย combination ของ regular expression และการเปรียบเทียบของตัวกระทำ

การใช้ภาษาระดับสูงเช่น C หรือ PASCAL หรือแม้แต่ภาษาแอสเซมบลีเพื่อจัดการเกี่ยวกับข้อมูลของการเปลี่ยนรูปแบบข้อมูล การตรวจความถูกต้องและการค้นหาภายในข้อมูล จะต้องใช้เวลาเขียนโปรแกรมนาน แต่ถ้าใช้ภาษา AWK จะเขียนได้สั้นมากเพียง 2-3 บรรทัดเท่านั้น ซึ่งหากมีเครื่องมือ AWK นี้แล้วจะช่วยให้การพัฒนาโปรแกรม (ไม่ว่าจะเขียนด้วยภาษาอะไร) ทำได้ง่ายขึ้น ตัวอย่างของเครื่องมือนี้ได้แก่ Poly-AWK ของ Polytron, MKS AWK ของ Mortice Kern System

สรุป

เครื่องมือสำหรับนักพัฒนาซอฟต์แวร์ในปัจจุบันมีอยู่มากมายหลายประเภทด้วยกัน ทุก ๆ ประเภทก็มีประโยชน์น่าใช้ทั้งนั้น ในบ้านเรายังไม่ค่อยมีการนำเครื่องมือเหล่านี้มาใช้เท่าไรนัก อาจมีสาเหตุหลายอย่างด้วยกัน อันแรกคิดว่าเป็นเพราะอุตสาหกรรมซอฟต์แวร์ในบ้านเรายังไม่เกิด ไม่มีการเขียนซอฟต์แวร์ขนาดใหญ่ ๆ เพื่อนำมาใช้งาน ส่วนใหญ่จะอาศัยจากต่างประเทศและถ้าต้องใช้ภาษาไทยจึงค่อยมาทำการแก้ไขดัดแปลงภายหลัง สาเหตุต่อไปอาจเนื่องจากวงการคอมพิวเตอร์ในบ้านเรายังอยู่ในวงแคบ คือยังไม่มีการใช้งานอย่างกว้างขวาง ระบบจัดการระบบงาน UNIX ที่นิยมใช้กันมากในต่างประเทศบนเครื่องที่สูงขึ้นกว่าไมโครคอมพิวเตอร์ยังไม่ค่อยมีในบ้านเราเลย ซึ่งในเครื่องที่ใช้ระบบ UNIX นี้มีเครื่องมือช่วยงานอยู่มาก เมื่อเราไม่รู้จักระบบ UNIX เราก็เลยพลอยไม่รู้จักเครื่องมือด้วย

แต่เป็นที่น่ายินดี ที่หลักการหรือเครื่องมือของ UNIX ก็มีการดัดแปลงนำมาใช้บนเครื่องไมโครคอมพิวเตอร์แล้ว คิดว่าเครื่องมือใหม่หลาย ๆ ตัวคงจะเข้ามาในบ้านเราบ้างในอนาคต ขณะเดียวกันโครงการเขียนซอฟต์แวร์ขนาดโต ๆ ก็เริ่มมีขึ้นมาบ้างแล้วในบ้านเรา อีกไม่นานเกินรอเราคงจะมีซอฟต์แวร์ของไทย ๆ เราเองใช้ ใหม่ ๆ อาจยังไม่ดีเทียบเท่าของต่างประเทศ แต่เชื่อว่าเราจะวิ่งตามเขาทันได้ในวันข้างหน้า ดีกว่าไม่เริ่มต้นเสียเลย